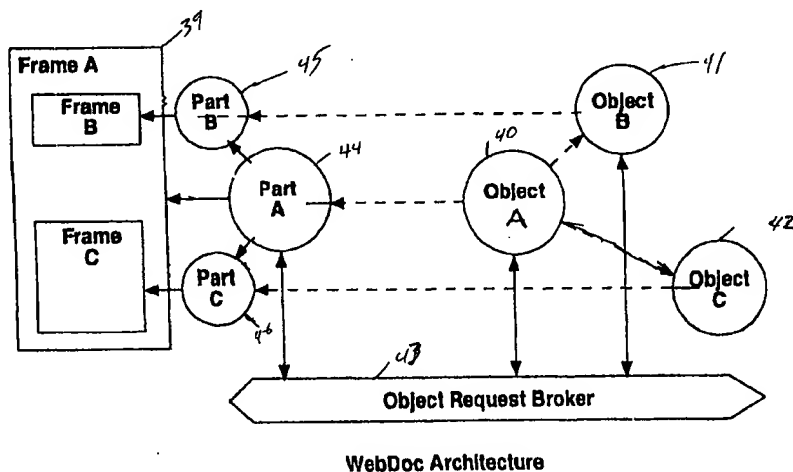




INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification ⁶ : G06F 17/30	A1	(11) International Publication Number: WO 98/21671 (43) International Publication Date: 22 May 1998 (22.05.98)
(21) International Application Number: PCT/US97/20393 (22) International Filing Date: 7 November 1997 (07.11.97) (30) Priority Data: 08/747,793 14 November 1996 (14.11.96) US (71) Applicant: TRITEAL CORPORATION [US/US]; 2011 Palomar Airport Road, Carlsbad, CA 92009 (US). (72) Inventor: KRAFT, Timothy, T.; 143 Sherry Lane, Del Mar, CA 92014 (US). (74) Agent: LAND, John; Fish & Richardson P.C., Suite 1400, 4225 Executive Square, La Jolla, CA 92037 (US).		(81) Designated States: JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE). Published <i>With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.</i>

(54) Title: DISTRIBUTED DOCUMENT PROCESSING THROUGH AN OBJECT REQUEST BROKER AND A RECEPTIONIST OBJECT



(57) Abstract

A system and method for an open GUI framework and distributed object server architecture using a URL-based approach to designating and accessing resources over the Internet. The resulting document is not only open, but also Internet distributed. A document is defined as the content of one or more distributed objects (41, 42), all linked to a receptionist object (40) directly or via an object request broker (43), and each of which is dynamically referenced in a "WebDoc" (39). Upon access via a URL to a server controlling the receptionist object, the WebDoc is assembled and transferred to a client viewer, where the object content is rendered as frames by parts that correspond to each object type. The results is a rendered document. In one embodiment, the parts may be linked back to corresponding objects, indirectly or via the object request broker, to maintain update information in the objects.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

DISTRIBUTED DOCUMENT PROCESSING THROUGH AN OBJECT REQUEST BROKER AND A RECEPTIONIST OBJECT

BACKGROUND OF THE INVENTION

1. *Field of the Invention*

5 This invention relates to network systems and object oriented software technology, and more particularly to a system and method for distributing objects and user interfaces over a network system.

2. *Description of Related Art*

10 With the advent of the Internet and object oriented programming, a number of different client/server and object oriented architectures have developed. For example, FIGURE 1 shows a browser/server architecture. A browser 1, typically on a client computer system, accesses a server 2, typically at a remote site, by using a universal resource locator (URL) 3 to locate the server 2 and a corresponding resource 4 on the server 2. This resource 4 is typically a hypertext markup language (HTML) file that contains URLs to other
15 resources 5 on the same server 2 or on other servers on the network. The referenced resource 4 is accessed by the server 2 and transmitted to the browser 1 using a conventional network protocol such as the TCP/IP protocol. The browser 1 renders the resource 4 as a viewable display object 4'. The viewable display object 4' may contain URLs 6 to
other resources 5.

20 The hypertext links (URLs) among HTML files distributed over the Internet define what is commonly called the world-wide web (WWW). The browser 1 is the user's means for "navigating" the WWW through URL selection. The user can "get" resources from distributed servers, and with the right permissions can "post" resources to distributed servers. This latter capability is typically reserved for one person called the "website
25 master".

Another model of object interaction over the Internet involves the JAVA computer programming language. With the introduction of JAVA, it is possible now to bring small applications called "applets" from a server to a browser to "run" with information rendered from an HTML file. Because JAVA code compiles to an intermediate
5 host-machine independent code, the compiled JAVA applets can run on any host computer that is executing a JAVA virtual machine (JVM). Using JAVA applets, entire applications can be downloaded by a user and run. (The difference between an applet and a full application is that the launcher for an applet enforces stricter security requirements that limits what the applet can access on client and host computer systems.)

10 Yet another object oriented architecture used in a network system is shown in FIGURE 2, which is a diagram of a distributed object request broker (ORB). In this architecture, each object is a potential Internet server object that performs a unit of work for another object that acts as a client object. Client objects can likewise act as server objects to other client objects. Objects are connected by means of the ORB 9, which serves as a message
15 delivery mechanism among all objects. Objects can locate and connect to each other via services performed by the ORB 9. In conventional implementations, communications between objects are abstracted by use of a proxy 10 and skeleton 11. The proxy 10 appears to a client object 12 to be the actual server object 13. That is, the proxy 10 links to the client object 12 in the same manner as if the server object 13 was directly linked to
20 the client object 12. On the server side, the skeleton 11 links to the server object 13 as a "stub" client, in a similar manner. The proxy 10 and skeleton 11 communicate via an interobject communications protocol supported by the ORB 9. Thus, the client object 12 appears to communicate over a direct communications link 14 with the server object 13. Similarly, the proxy 10 appears to directly communicate over communications link 15
25 with the skeleton 11. In reality, all objects communicate through the ORB 9. The proxy 10 and skeleton 11 representations can be generated automatically from the corresponding original object source code, or from an intermediate design language (IDL) description of the corresponding object.

While the ORB 9 in FIGURE 2 is shown as a unified structure, conventionally, the ORB 9 is a network of ORBs executing on different computer systems but interconnected via the Internet. The interconnected ORBs form a virtual software bus that allows objects to be distributed and potentially run anywhere on the Internet.

- 5 FIGURE 3 shows yet another diagram of a software architecture and implementation for constructing compound documents from simple parts. This architecture is generally referred to as an "OpenDoc" architecture. Objects called "Parts" represent the content of a document through a tree hierarchy that defines the encapsulation of the compound Parts of the document. In the illustrated example, Part A is designated as the "Root" Part of the document Part tree, and defines the overall layout of the encapsulated Parts via a Frame 10 20. Frames represent the visual and geometry elements of the document content, and each Frame has a corresponding Part that will "render" or "paint" its content within the Frame. Thus, for example, Part B corresponds to Frame B and will paint its content into Frame B. Part C corresponds to Frame C and will paint its content into Frame C.
- 15 The Parts 20 and Frames 21 that form a document are stored in an object called a "Bento" container 22, which allows the individual elements to be accessed and cataloged independently from the rendered document. All of the elements of a document can be accessed and rendered by various viewers and editors. Thus, the document is considered to be "open". Although a number of other elements make up the OpenDoc architecture, 20 the Parts 20 and Frames 21, and the interconnection among these objects, form the essential description of an arbitrary document's content.

While the architectures described above provide a fair degree of flexibility in referencing and communicating between objects in a network environment, they still lack the ability to define an open GUI Framework and distributed object server architecture that uses the 25 prevalent URL-based approach to designating and accessing resources over the Internet. The present invention provides such an architecture.

SUMMARY OF THE INVENTION

The invention comprises a system and method for an open GUI Framework and distributed object server architecture using a URL-based approach to designating and accessing resources over the Internet. The resulting document is not only open, but also
5 Internet distributed.

More particularly, a document is defined as the content of one or more distributed objects, all linked to a receptionist object directly or via an object request broker, and each of which is dynamically referenced in a "WebDoc". Upon access via a URL to a server controlling the receptionist object, the WebDoc is assembled and transferred to a client
10 viewer, where the object content is rendered as Frames by Parts that correspond to each object type. The result is a rendered document. In one embodiment, the Parts may be linked back to corresponding objects, indirectly or via the object request broker, to maintain update information in the objects.

The details of the preferred embodiment of the invention are set forth in the accompanying drawings and the description below. Once the details of the invention are known,
15 numerous additional innovations and changes will become obvious to one skilled in the art.

BRIEF DESCRIPTION OF THE DRAWINGS

FIGURE 1 is diagram of a prior art browser/server architecture.

FIGURE 2 is diagram of a prior art distributed object request broker architecture.

FIGURE 3 is diagram of a prior art OpenDoc architecture.

5 FIGURE 4 is a diagram showing the basic WebDoc document architecture of the invention.

FIGURE 5 is a diagram showing the mechanism for retrieving and rendering a WebDoc using a client viewer.

10 FIGURE 6 is a diagram showing the format of a WebDoc constructed to represent the content of an object.

FIGURE 7 is a diagram showing an alternative format of a WebDoc constructed to represent the content of an object.

FIGURE 8 is a screen-shot of a example WebDoc, showing a desktop document which loads WebDocs for other universal resource locators.

15 Like reference numbers and designations in the various drawings indicate like elements.

DETAILED DESCRIPTION OF THE INVENTION

Throughout this description, the preferred embodiment and examples shown should be considered as exemplars, rather than as limitations on the invention.

This invention is related to the inventions entitled "Method for Displaying Multiple Visual Styles" (U.S. patent application serial no. _____) and "URL Login" (U.S. patent application serial no. _____), filed on the same date as the present patent application.

Overview

The invention comprises a system and method for an open GUI Framework and distributed object server architecture using a URL-based approach to designating and accessing resources over the Internet. The resulting document is not only open, but also Internet distributed. For ease of reference, this document will be referred to as a "WebDoc".

Advantages and uses for WebDocs include the following:

- (1) A WebDoc can be used to bring up a desktop session on a Network Computer and save the session initiated by a user.
- (2) User-constructed WebDocs can be presented on a variety of computer systems or display devices, such as pagers, personal digital assistants, Network Computers, personal computers, workstations, and mainframes.
- (3) The preferred embodiment of the invention incorporates current mail format standards. Accordingly, little or no modifications are necessary for current Internet content handlers. Indeed, any Multipurpose Internet Mail Extensions (MIME) compliant mailer application can handle WebDoc content at some level.
- (4) Because WebDocs are flat collections of "bodyparts" (as opposed to nested compound structures), WebDoc documents can be streamed and processed in parts, constructed from multiple streams, and processed in parallel.

- (5) In the preferred embodiment, a WebDoc is compliant with all current Internet standards and supports current encryption and signature methodologies.
- (6) A WebDoc can be routed through current MIME-compliant Internet gateways, including email applications providing multiple methods for transmitting and receiving WebDocs.
- (7) A WebDoc can be used to store data content, user interface descriptions, persistent states for objects, and even object code, all in one document.
- (8) Messages between servers are supported, so that the same WebDoc support functions can support interobject communications over a distributed unsecured Internet.
- (9) The invention provides a standard framework for distributed object architectures to interact with a network computer desktop.
- (10) The supporting WebDoc system of objects can be distributed and the functions of data collection, document construction and collection, transmission, and display, can be performed anywhere on the Internet.
- (11) The supporting WebDoc system in the preferred embodiment can process and display current MIME-encoded messages (including MIME HTML mail messages) in a "viewer" selected and constructed by the user, as well as allow content to be incorporated into the user's own compound documents.

WebDoc Architecture

The architecture of a WebDoc system includes a WebDoc document format and software components with characteristics to receive, process, and appropriately interpret the content of a WebDoc. FIGURE 4 is a diagram showing the basic WebDoc architecture. This general architecture allows remote location for objects comprising a WebDoc, with a robust mechanism for accessing and rendering all of such objects.

A final WebDoc 39 is constructed by a receptionist object 40 that communicates with other content objects 41, 42 through an object request broker 43 that is responsible for bringing the various elements of the final WebDoc 39 together. Every object 41, 42 that

represents the content of the receptionist object 40 has a corresponding WebDoc. These other objects 41, 42 may be internal to the receptionist object 40 (such as object C) or may be accessible to the receptionist object 40 through the ORB 43 (such as object B). In the preferred embodiment, all objects 41, 42 are linked to the receptionist object by means of URLs.

The receptionist object 40 provides a standard server interface that provides access to one or more services performed by other encapsulated objects. In the preferred embodiment, the receptionist object 40 provides functions for authenticating user accesses to the service, as well as constructing the appropriate WebDoc for the requested service. The content of a receptionist object 40 does not have to be physically located within the receptionist object 40, but can itself be distributed on other Internet nodes.

The receptionist object 40 corresponds to the Root Part 44 of the final WebDoc 39. Other Parts 45, 46 contained within the Root Part 44 similarly correspond to other objects 41, 42 encapsulated by the receptionist object 40. The receptionist object 40 fetches the content (WebDocs) of all of the encapsulated objects 41, 42, and transmits that content to the Root Part 44 and its sub-Parts 45, 46. As in the OpenDoc standard, the Parts 44, 45, 46 define corresponding Frames, which together form the final WebDoc 39. That is, each Frame has a corresponding Part that will "render" or "paint" its content within the Frame.

In the illustrated example, the receptionist object 40 may directly get the content of object C in the form of a WebDoc which defines the content of corresponding Part C. The content of object B may be fetched via the ORB 43.

As an option, each Part 44, 45, 46, as an object, can form a direct connection through the ORB 43 to its corresponding object 40, 41, 42 in order to maintain current content in the associated object (*e.g.*, saving user input). Thus, each Part can keep its content up-to-date using its own access methods. In the preferred embodiment, the connection through the

ORB 43 is only through the Root Part 44. The Root Part 44 coordinates communications between its embedded Parts 45, 46 and their corresponding objects 41, 42.

In the preferred embodiment, every Part 44, 45, 46 that makes up the final WebDoc 39 has a document unique Content Identifier (CID) and a network unique universal-object
5 resource indicator (URI), the use of which is described below.

FIGURE 5 is a diagram showing the mechanism for retrieving and rendering a final WebDoc 49 using a client-resident viewer 50. The viewer 50 supplies a URL 51 to a launcher 52. The URL 51 is used to initiate an access to a remote receptionist object 53 via a server 54. The launcher 52 coordinates access to the server 54 via an Internet link
10 55, as well as the construction of the final WebDoc 49 on a client computer system (node).

The server 54 directly accesses a WebDoc file from a receptionist object 53, or requests the WebDoc file from the receptionist object 53, depending upon the URL Internet access protocol being used over the Internet link 55. In either case, the WebDoc file is the same
15 and is independent of the access method of the URL Internet access protocol. The receptionist object 53 fetches ("gets") the content of the WebDoc file (in WebDoc format) from all encapsulated objects 56a-56d, either directly or through an ORB 57.

The accessed WebDoc file 55 is received by the launcher 52 and viewer 50, which select an appropriate WebDoc handler parser and initiate the various document Parts for the
20 contents of the WebDoc file 55. As in FIGURE 4, a Root Part 58 and sub-Parts 59a-59b correspond to each object. The Parts define corresponding Frames, which together form the final WebDoc 49. The Root Part 58 and its constituent sub-Parts are rendered on the viewer 50, with each Part rendering its content within a corresponding Frame.

Again, as an option, each Part, as an object, can form a direct connection 60 through the
25 ORB 57 to its corresponding object in order to maintain current content. Thus, each Part

can keep its saved content up-to-date using its own access methods. In the preferred embodiment, each Part handles "saving" or "maintaining" its internal content in its corresponding object. Also, the connection 60 to the ORB 57 is preferably only through the Root Part 58.

5 An alternative approach is to not directly connect Parts with their corresponding objects via a connection 60 to the ORB 57. This approach may be required for security reasons or by resource limitations on the client node. An alternative mechanism can be used that omits the connection 60 to the ORB 57. In this mechanism, the viewer 50 and/or launcher 52 saves the current state of each Part of the final WebDoc 49 in a WebDoc file that has
10 the exact same form as the WebDoc file that launched the Part (as an option, only Parts that have changed are saved). Using the original URL 51, this changed WebDoc is transferred to the remote server 54 for posting. Depending on the URL Internet access protocol being used, the changed WebDoc is saved as a file on the server 54 or is transferred to the receptionist object 53 for update processing. In the case of a file save,
15 the server 54 determines how the updated information is merged into the original WebDoc. In the case of direct object access via the receptionist object 53, the receptionist object 53 processes the changed WebDoc to either update the content of the receptionist object 53 or to distribute portions of the WebDoc to sub-objects for similar object updating.

20 **WebDoc Document Format**

In the preferred embodiment of the invention, the content of a WebDoc is a MIME-encoded multipart message. The description above describes how these messages can be used to transport all content among distributed objects via the WWW.

25 The content of any object can be given as either parameter name/value pairs, or as internal formatted content, such as HTML, JPEG, QuickTime, binary, *etc.* An important aspect of the WebDoc concept is the ability to format complex content from elements that can be readily understood across the Internet, as opposed to developing special proprietary

content formats requiring appreciably more development and support. Using object oriented programming (OOP), complex objects are constructed from simpler objects so that the corresponding content of the object is a compound document comprising multiple parts.

5 FIGURE 6 is a diagram showing the format of a WebDoc constructed to represent the content of an object. The document is formatted as a multipart/related "message" consisting of multiple content bodyparts. A bodypart (typically the first) is designated as the root part 70. The root part 70 contains a description of the layout of the other bodyparts and references these bodyparts through unique content identifiers (CIDs).

10 Each object generates a corresponding bodypart in the WebDoc that describes the content and location of the object, preferably using a Standard Generalized Markup Language (SGML) formatted content, such as HTML, that defines content in relation to other objects. This description translates to the layout of Frames in a final document. Thus, the receptionist object (Object A in FIGURE 6) describes its content as the root part 70 of the
15 WebDoc. Each corresponding Part (here, Part A) takes this description and renders the content. Similarly, each encapsulated object (here, Objects a-d) generates bodyparts in the WebDoc that define the content of those objects. Each corresponding part (Parts a-d) renders such content, and thus reconstructs the document from its constituent parts, gathered from across the Internet.

20 In the preferred embodiment, each bodypart also maintains a universal resource identifier (URI) to its corresponding object source, which is the "content location" for the bodypart. The URI preferably takes the form of a URL.

Handlers for the bodyparts are selected primarily upon the content-type of each bodypart. For example, bodyparts having text content are handled by a text-viewing Part, bodyparts
25 having image content are handled by an image-viewing Part, *etc.* As a result, the Part

viewers can be considered to be part of the document viewer (e.g., viewer 50 in FIGURE 5) and not necessarily transported as part of the WebDoc.

On the other hand, a Part's viewer may have configurable attributes which allows the Part to be customized to perform specific functions on behalf of the corresponding object.

5 Such configuration attributes are passed to a Part's viewer as parameter name/value pairs which are preferably incorporated as headers in the WebDoc bodypart that precedes the content of the message.

FIGURE 7 is a diagram showing an alternative format of a WebDoc constructed to represent the content of an object. This format can be used when the layout of the constituent parts is either implicit to a Part's viewer, or the Part's viewer can be
10 configured through parameter name/value pairs. No special root body part is necessary because the main document header 80 provides additional content-viewer parameters and Part viewer parameters to configure the Part content from the document content.

For a compound document, the content type is multipart/related. The document parts are
15 constructed as before (FIGURE 6) from the constituent object contents. There may be additional parameters added to aid in the layout of an embedded document part in relation to the root document part and the other parts in the document.

An advantage of formatting WebDocs in this manner is that the parts can still be rendered even if the target viewer is not aware of all of the attributes of the document, or lacks all
20 of the special Part viewers or viewer capabilities. Thus, various classes of viewers can view the same WebDoc because the WebDoc is MIME formatted and encoded. On the other hand, viewers and their event handlers can be exactly configured to render a WebDoc and its active elements exactly as intended with Part viewers that are WebDoc "aware".

Example

FIGURE 8 is a screen-shot of a example WebDoc, showing a desktop document which loads WebDocs for other universal resource locators. More particularly, FIGURE 8 illustrates a Network Computer desktop constructed entirely from a combination of WebDoc components. The desktop 100 is a WebDoc document consisting of multiple work areas, with one of the work areas shown in full frame. The desktop 100 defines a framework for overlapping windows 102, 104, 106, 108, 110, each comprising a WebDoc, and icons 112, 114. The icons and windows can be moved and resized using a mouse typical of other graphical desktops (*e.g.*, X-Windows, Windows95, OS2, MacOS, *etc.*).

Window 102 is a Graphical Workspace Manager which is constructed as a “thumbnail” view of the desktop 100 Part. The same Part is used to “paint” another view of a WebDoc in a separate Frame. Any WebDoc Part can provide a thumbnail view which is a reduced image of the WebDoc Part’s “full view”. Yet another view (*i.e.*, Frame) 116 of the desktop 100 is embedded into a “front panel” window 106. All views are coupled so that changes in one view will be reflected in all other views (frames). In the preferred embodiment, the Graphical Workspace Manager 102 is viewed within a Frame which is itself constructed from WebDoc Parts for the title bar and layout of the desktop thumbnail views.

Window 104 is an Internet Navigator constructed from a WebDoc document, showing the contents of a receptionist object that contains files. Changes in this view are transmitted to the receptionist object to save. The receptionist object likewise keeps this view “current” via either WebDoc messaging or an ORB connection.

Window 106 is a front panel of the type used in the industry standard Common Desktop Environment (CDE), but constructed and configured by a WebDoc document and Parts. The icons within the front panel represent services on the Internet that a user can activate

by a mouse click. This action results in loading and displaying a WebDoc document within a viewer.

A subpanel 118 is raised from the front panel 106 when the user presses an arrow button 120. The subpanel 118 shown contains an "icon view" of WebDoc Parts. These icons can
5 be used to raise a full view of the corresponding WebDoc. These icons can also be active drop zones for other WebDocs or for URLs which the WebDoc Part can process.

Icon 114 is an "icon view" of a WebDoc. Like the thumbnail view shown in window 102, the icon view is another Frame for a Part. When a Part is depicted in an icon view, it is not necessary to retrieve the entire corresponding WebDoc contents. Typically, what is
10 required to support a WebDoc icon view is the URL to the corresponding full view content and Part event handler for processing URLs that are dropped on the displayed icon 114.

Frame 116 is another thumbnail view of the desktop 100, embedded into the front panel 106. Within the Frame 116 are thumbnail views of several workspaces 122 (shown with
15 miniature views of their backdrop patterns). Each workspace 122 also depicts thumbnail views of WebDocs 124 and icons 126 contained within the workspace 122. Each thumbnail view of a WebDoc has the same functionality as the corresponding full view of the WebDoc. That is, the thumbnail view can receive and respond to mouse and keyboard events in the same manner as the full view, since the exact same Part supports
20 both views. Although the same functionality can be implemented in a conventional architecture (*e.g.*, CDE), the amount of code required to implement such functionality is greatly reduced and uniformity is greatly enhanced using WebDocs.

Window 108 shows an applet displayed without modification within an applet viewer constructed from WebDoc Parts. Because applets can be viewed as Parts, a user can
25 incorporate applets (developed for the Internet) into the user's WebDocs. In a similar

fashion, OpenDoc parts, Netscape Navigator “plugin” modules, and Microsoft ActiveX components can be incorporated into the desktop or user documents.

Window 110 shows a text editor constructed from WebDoc Parts. By changing the WebDoc user interface for the viewer, a user can configure the editor to perform desired functions, including incorporating Parts for spell checking or text formatting. The Parts for this function would describe a viewer for handling, in this case, text. The WebDoc generated by the viewer/editor would be simply the text content (in this case). In general, there are levels of WebDoc content: (a) only the contents that have been changed; (b) the original contents with any modifications; (c) contents with the UIL (user interface language) descriptions of the viewer; (d) the contents, viewer UIL, and binary codes.

Icon 112 is an icon view of an application that is not a WebDoc. The application is referred to as a “legacy” application, which is any application developed outside of the WebDoc paradigms. In the preferred embodiment, WebDoc references and links to legacy applications are via URLs. In the preferred embodiment, the WebDoc-defined desktop can activate (display) legacy applications in their own window, and through the application program interface (API) for the underlying operating system (*e.g.*, Windows95) transfer WebDoc contents to the application. In specific implementations, “drag and drop” functionality between legacy and WebDoc components can be supported.

General Implementation

The invention may be implemented in hardware or software, or a combination of both. However, preferably, the invention is implemented in computer programs executing on programmable computers each comprising at least one processor, at least one data storage system (including volatile and non-volatile memory and/or storage elements, any of which may be local or accessible via a network), at least one input device, and at least one output device. Program code is applied to input data to perform the functions described herein and generate output information. The output information is applied to one or more output devices, in known fashion.

Each program is preferably implemented in a high level procedural or object oriented programming language to communicate with a computer system, although the programs can be implemented in assembly or machine language, if desired. In any case, the language may be a compiled or interpreted language. However, in the preferred
5 embodiment, the invention is implemented in the JAVA programming language for use in conjunction with any host that has a JVM.

Each such computer program is preferably stored on a storage media or device (*e.g.*, ROM or magnetic diskette) readable by a general or special purpose programmable computer, for configuring and operating a local or remote computer when the storage media or
10 device is read by the computer to perform the procedures described herein. The inventive system may also be considered to be implemented as a computer-readable storage medium, configured with a computer program, where the storage medium so configured causes a computer to operate in a specific and predefined manner to perform the functions described herein.

15 A number of embodiments of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, it is to be understood that the invention is not to be limited by the specific illustrated embodiment, but only by the scope of the appended claims.

CLAIMS

What is claimed is:

1. A method for defining a graphical document in a distributed network computer system having at least one network server and at least one client, comprising the steps of:
 - (a) defining at least one distributed content object having content to be viewed in the graphical document;
 - (b) defining a receptionist object having content that includes at least one link to at least one such content object;
 - (c) coupling the receptionist object and at least one content object to an object request broker, the object request broker providing network-wide access and communication between the receptionist broker and each coupled content object;
 - (d) accessing, through the network server, the contents of each content object linked to the receptionist object;
 - (e) transmitting the accessed contents from the network server as a document definition file to viewer/launcher application executing on the client;
 - (f) rendering the accessed contents defined in the document definition file as frames of the graphical document.

2. The method of claim 1, wherein each link in the receptionist document is a universal resource locator.
3. The method of claim 1, further including the step of rendering each content object by a corresponding Part.
4. The method of claim 1, wherein each Part is linked back to a corresponding object, and further including the step of saving changes to a frame in the corresponding object.
5. The method of claim 4, wherein each Part is linked back to a corresponding object through the object request broker.
6. The method of claim 4, wherein each Part is linked back to a corresponding object through the receptionist object.
7. The method of claim 1, wherein the document definition file comprises at least one bodypart, and each bodypart describes the content and location of a corresponding object.

8. A system for defining a graphical document in a distributed network computer system having at least one network server and at least one client, comprising:
- (a) means for defining at least one distributed content object having content to be viewed in the graphical document;
 - 5 (b) means for defining a receptionist object having content that includes at least one link to at least one such content object;
 - (c) means for coupling the receptionist object and at least one content object to an object request broker, the object request broker providing network-wide access and communication between the receptionist broker and each
10 coupled content object;
 - (d) means for accessing, through the network server, the contents of each content object linked to the receptionist object;
 - (e) means for transmitting the accessed contents from the network server as a document definition file to viewer/launcher application executing on the
15 client;
 - (f) means for rendering the accessed contents defined in the document definition file as frames of the graphical document.

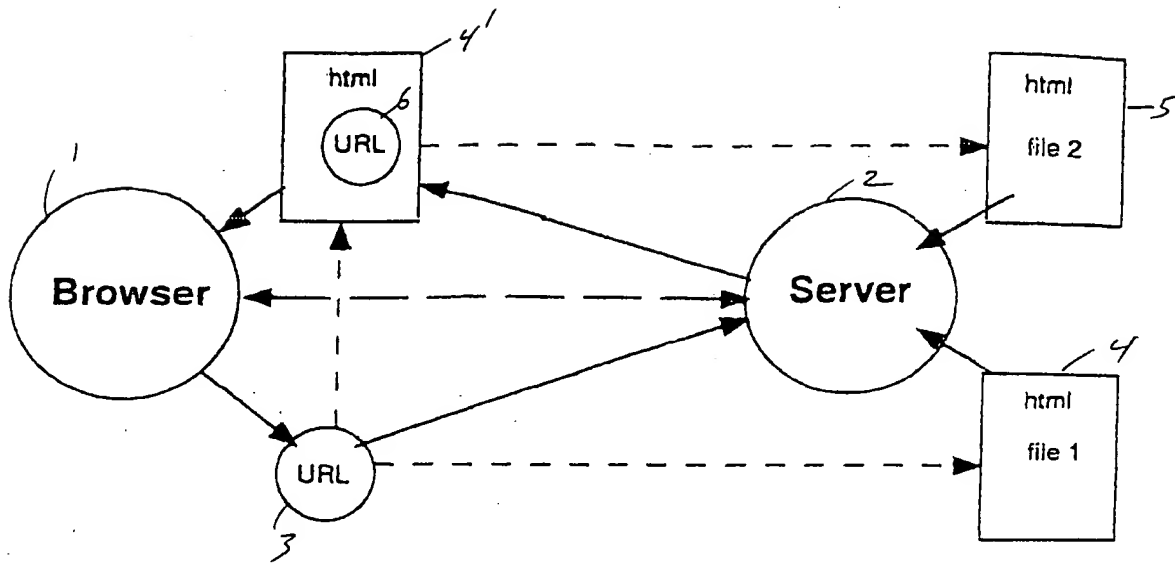
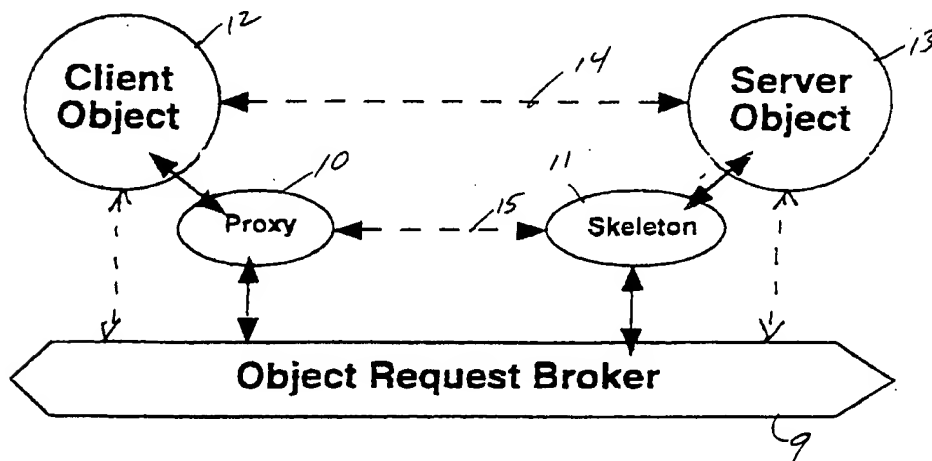
9. The system of claim 8, wherein each link in the receptionist document is a universal resource locator.
10. The system of claim 8, further including means for rendering each content object by a corresponding Part.
11. The system of claim 8, wherein each Part is linked back to a corresponding object, and further including means for saving changes to a frame in the corresponding object.
12. The system of claim 11, wherein each Part is linked back to a corresponding object through the object request broker.
13. The system of claim 11, wherein each Part is linked back to a corresponding object through the receptionist object.
14. The system of claim 8, wherein the document definition file comprises at least one bodypart, and each bodypart describes the content and location of a corresponding object.

15. A computer program, residing on a computer-readable medium, for defining a graphical document in a distributed network computer system having at least one network server and at least one client, the computer program comprising instructions for causing a computer to:

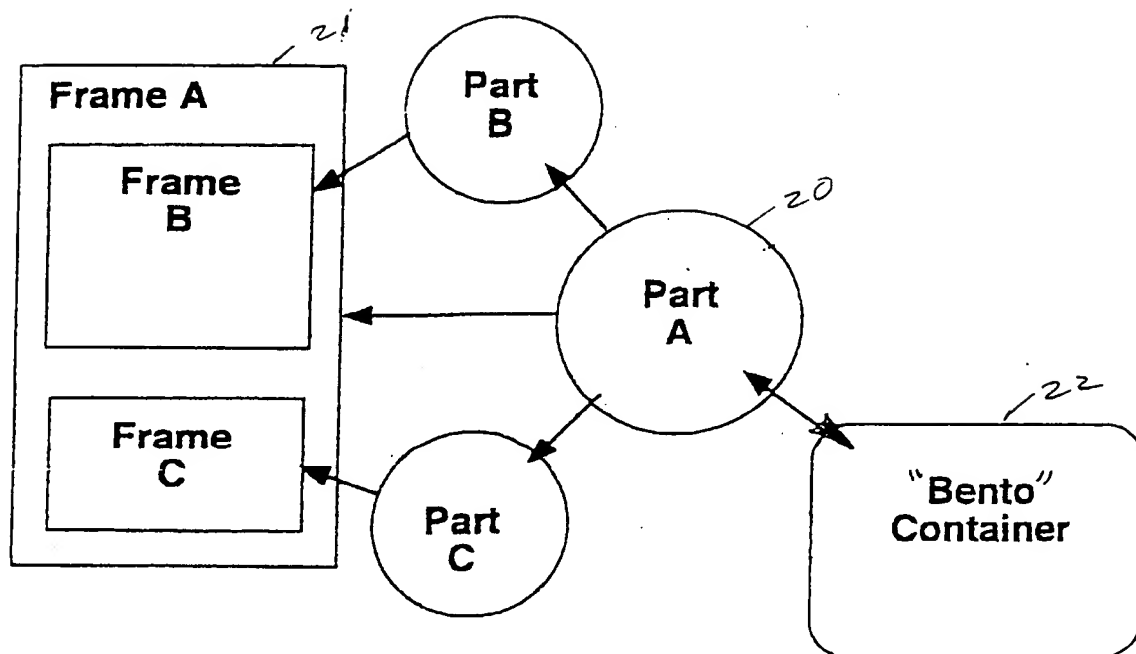
- 5 (a) define at least one distributed content object having content to be viewed in the graphical document;
- (b) define a receptionist object having content that includes at least one link to at least one such content object;
- (c) couple the receptionist object and at least one content object to an object
10 request broker, the object request broker providing network-wide access and communication between the receptionist broker and each coupled content object;
- (d) access, through the network server, the contents of each content object linked to the receptionist object;
- 15 (e) transmit the accessed contents from the network server as a document definition file to viewer/launcher application executing on the client;
- (f) render the accessed contents defined in the document definition file as frames of the graphical document.

16. The computer program of claim 15, wherein each link in the receptionist document is a universal resource locator.
17. The computer program of claim 15, further comprising instructions for causing a computer to render each content object by a corresponding Part.
18. The computer program of claim 15, wherein each Part is linked back to a corresponding object, and further comprising instructions for causing a computer to save changes to a frame in the corresponding object.
19. The computer program of claim 18, wherein each Part is linked back to a corresponding object through the object request broker.
20. The computer program of claim 18, wherein each Part is linked back to a corresponding object through the receptionist object.
21. The computer program of claim 15, wherein the document definition file comprises at least one bodypart, and each bodypart describes the content and location of a corresponding object.

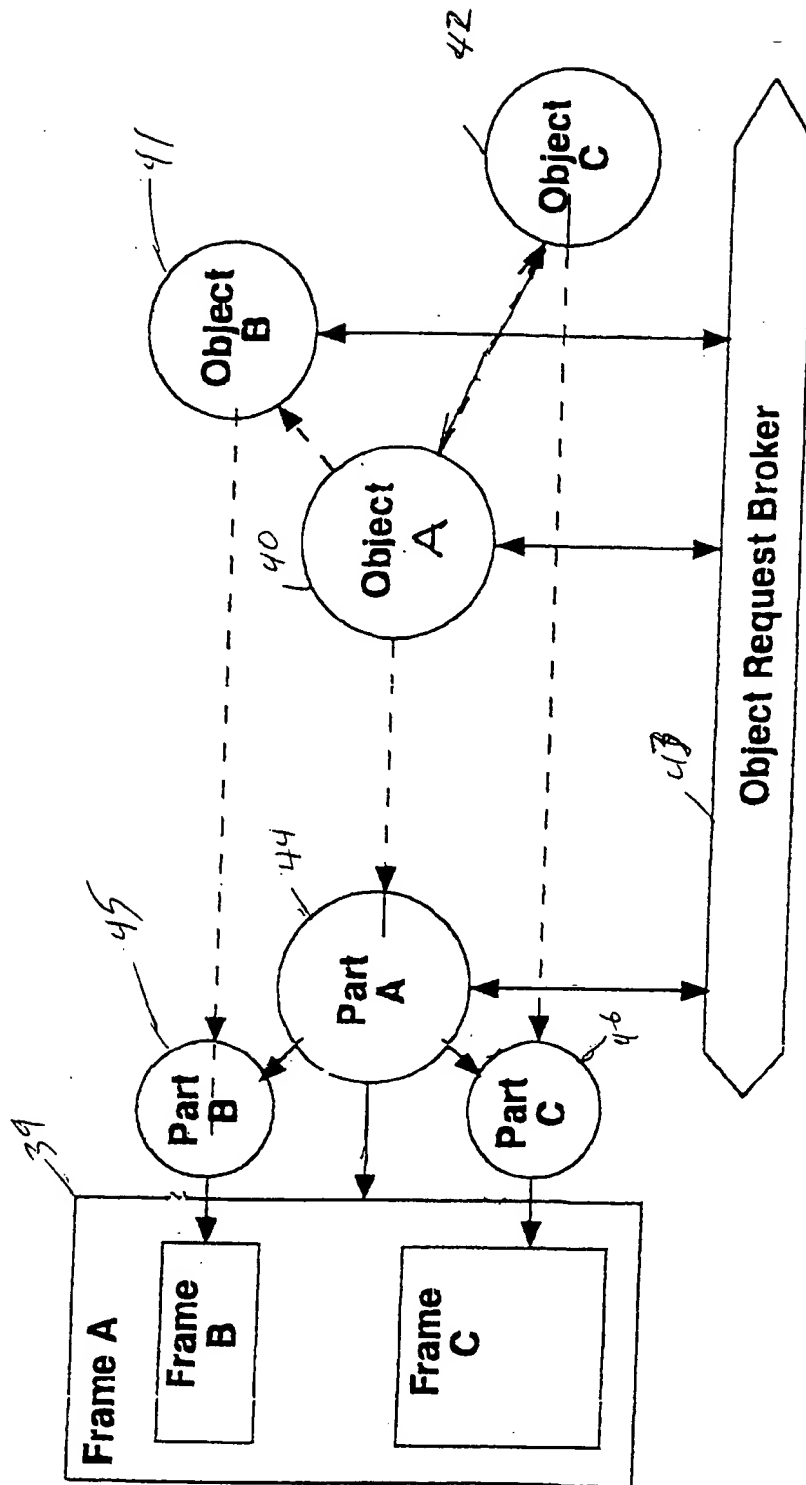
1/7

**Browser/Server***Fig. 1
(Prior Art)***Distributed Object Request Broker***Fig. 2
(Prior Art)*

2/7

**OpenDoc***Fig. 3
(Prior Art)*

3 / 7



WebDoc Architecture

Fig. 4

4/7

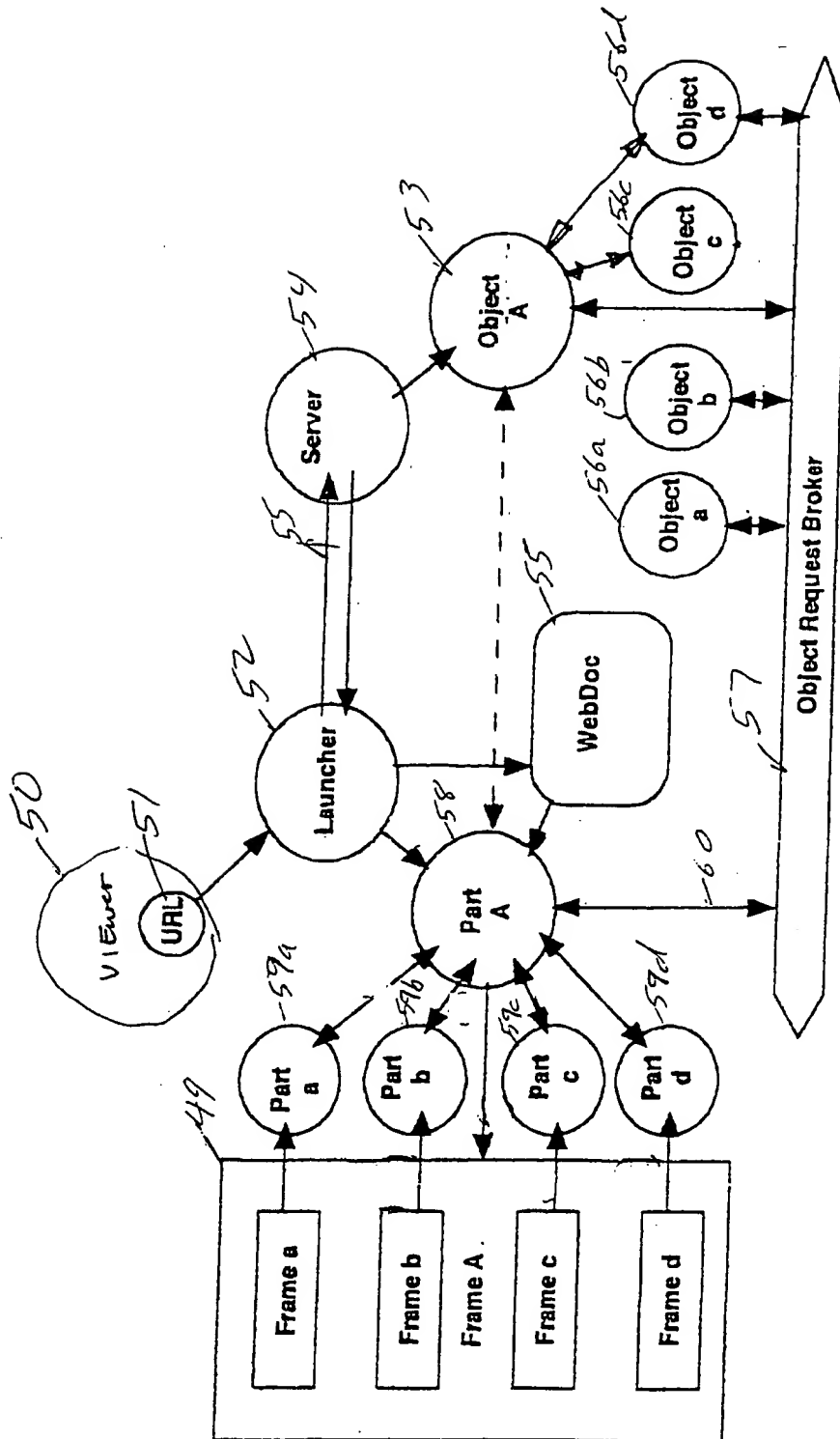
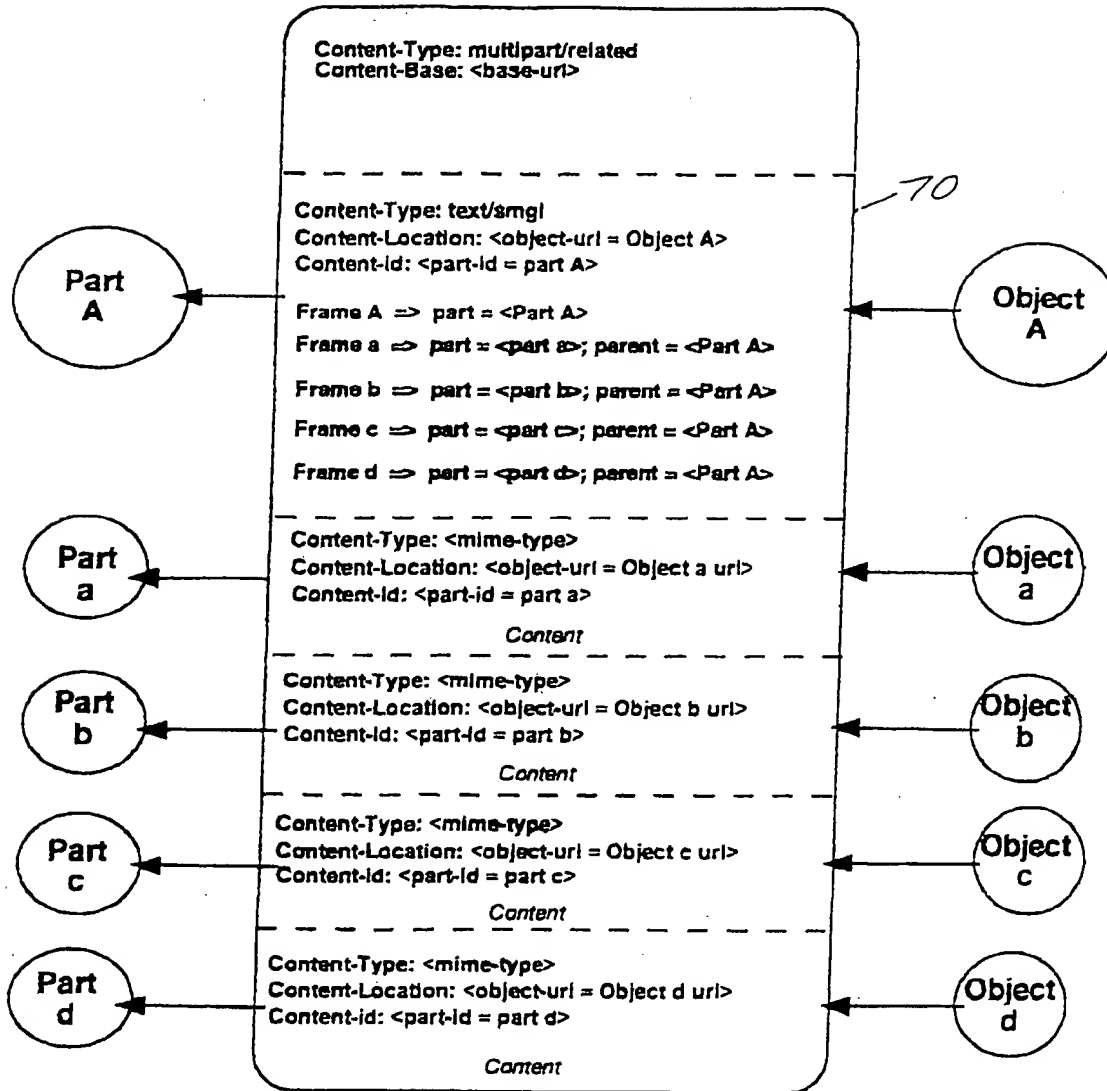


Fig. 5

5/7



WebDoc

Fig. 6

6/7

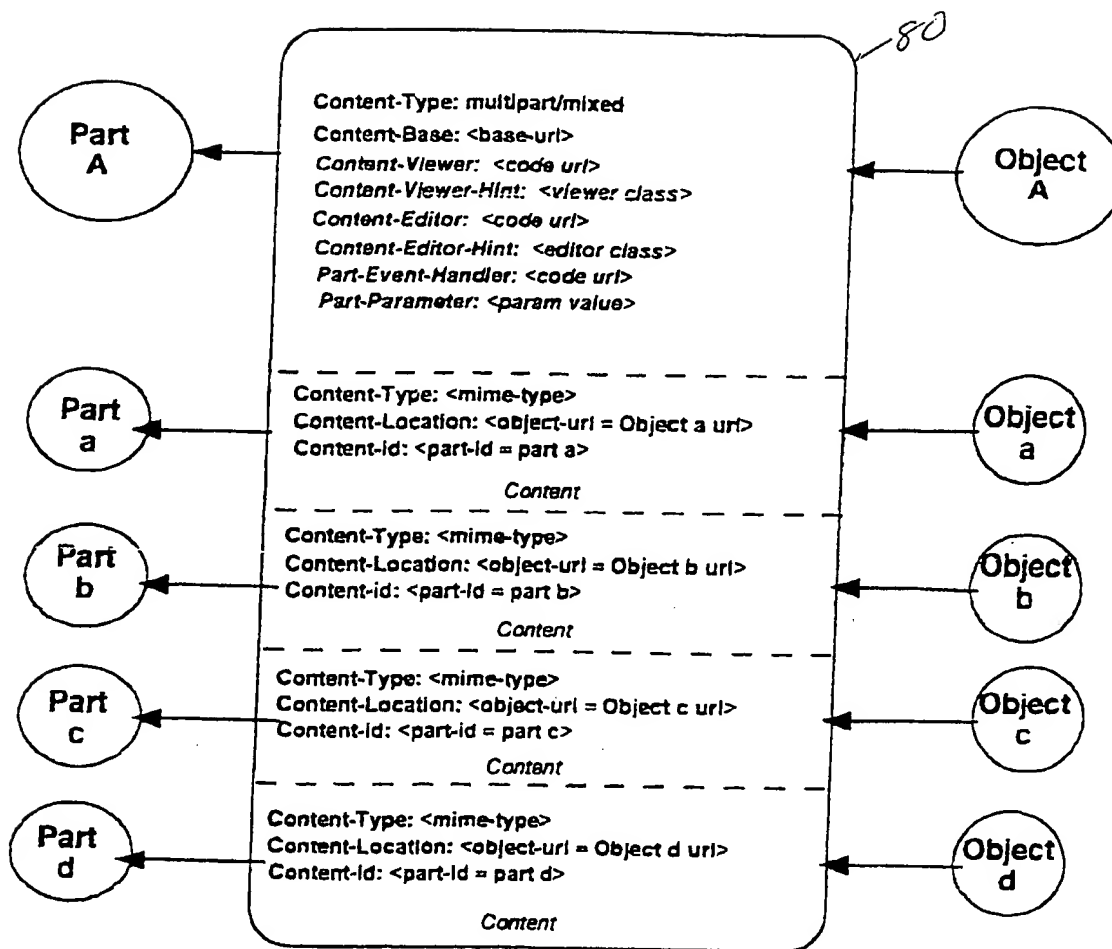
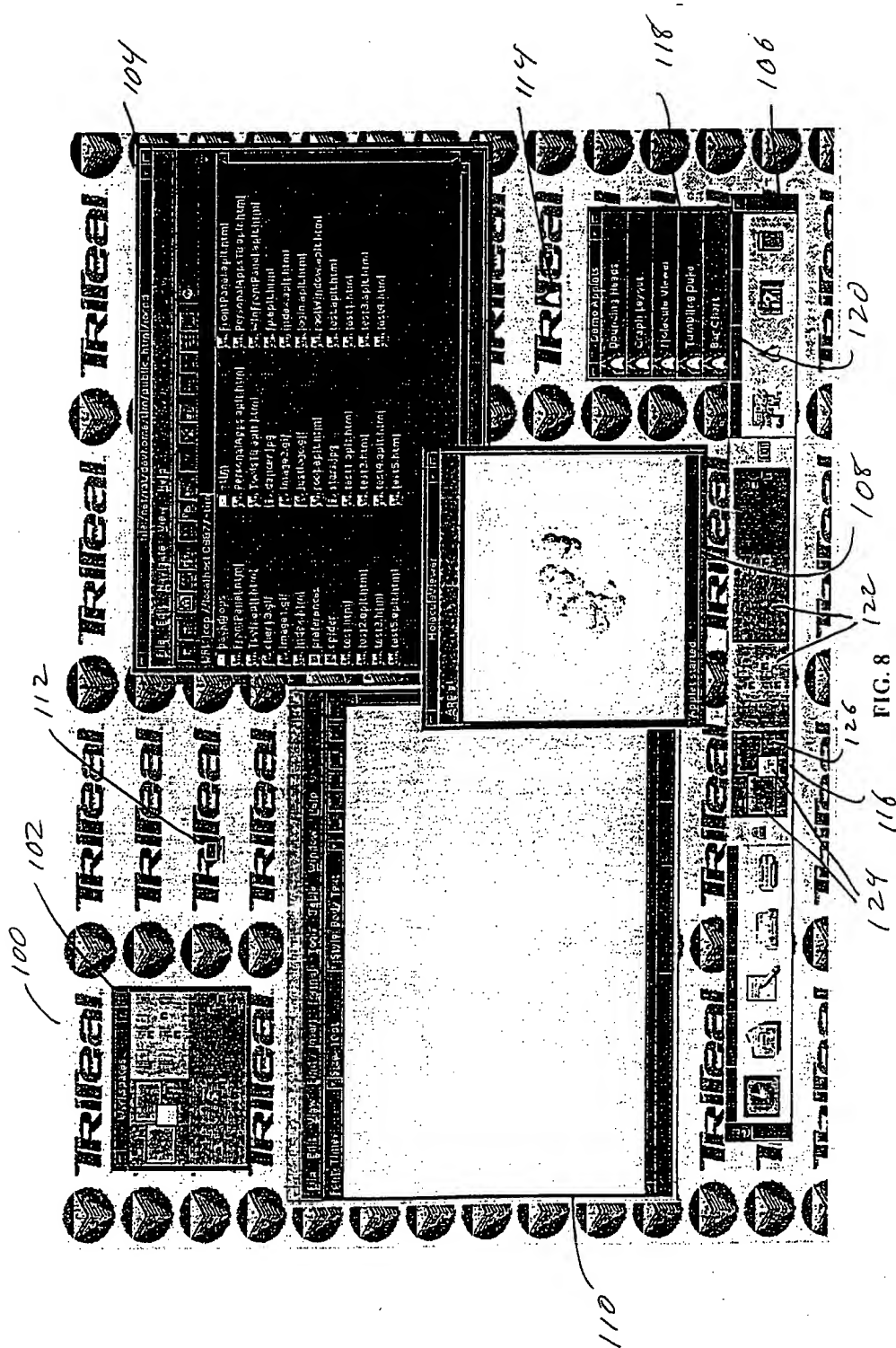


Fig. 7

7 / 7



INTERNATIONAL SEARCH REPORT

 International application No.
PCT/US97/20393

A. CLASSIFICATION OF SUBJECT MATTER		
IPC(6) : G06F 17/30		
US CL : 707/515, 10		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
U.S. : 707/515, 10		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)		
APS: search terms: receptionist object; object request broker; distributed; object; broke?; document; display; graphical; network; web; OpenDoc		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,E	US 5,701,484 A (ARTSY) 23 DECEMBER 1997, see entire document.	1-21
A,E	US 5,689,664 A (NARAYANAN et al.) 18 NOVEMBER 1997, see entire document.	1-21
A,P	US 5,664,208 A (PAVLEY et al.) 02 SEPTEMBER 1997, see entire document.	1-21
A	US 5,491,820 A (BELOVE et al.) 13 FEBRUARY 1996, see entire document.	1-21
A	US 5,481,715 A (HAMILTON et al.) 02 JANUARY 1996, see entire document.	1-21
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.		
* Special categories of cited documents:	*T*	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
A document defining the general state of the art which is not considered to be of particular relevance	*X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
E earlier document published on or after the international filing date	*Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
L document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*G*	document member of the same patent family
O document referring to an oral disclosure, use, exhibition or other means		
P document published prior to the international filing date but later than the priority date claimed		
Date of the actual completion of the international search	Date of mailing of the international search report	
13 FEBRUARY 1998	25 MAR 1998	
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer JOSEPH BURWELL	
Facsimile No. (703) 305-3230	Telephone No. (703) 305-3800	

Form PCT/ISA/210 (second sheet)(July 1992)*